PFJA-00065-US

日 本 国 特 許 庁
PATENT OFFICE
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日
Date of Application: ２０００年 ７月１３日

出 願 番 号
Application Number: 特願２０００－２１２４８２

出 願 人
Applicant (s): 富士通株式会社

CERTIFIED COPY OF
PRIORITY DOCUMENT

２０００年１２月２２日

特許庁長官
Commissioner,
Patent Office 及川耕造

出証番号 出証特２０００－３１０５６４９

【書類名】　特許願

【整理番号】　0051429

【提出日】　平成12年 7月13日

【あて先】　特許庁長官　殿

【国際特許分類】　G06F 15/00

【発明の名称】　拡大鍵生成装置および記録媒体

【請求項の数】　6

【発明者】

【住所又は居所】　神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内

【氏名】　下山　武司

【発明者】

【住所又は居所】　神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内

【氏名】　伊藤　孝一

【発明者】

【住所又は居所】　神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内

【氏名】　武仲　正彦

【発明者】

【住所又は居所】　神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内

【氏名】　鳥居　直哉

【発明者】

【住所又は居所】　神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内

【氏名】　矢嶋　純

【発明者】

【住所又は居所】　神奈川県川崎市中原区上小田中４丁目１番１号　富士通

株式会社内

【氏名】　屋並　仁史

【発明者】

【住所又は居所】　神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内

【氏名】　横山　和弘

【特許出願人】

【識別番号】　000005223

【氏名又は名称】　富士通株式会社

【代理人】

【識別番号】　100089141

【住所又は居所】　東京都目黒区平町１丁目２１番２０－６０３号

【弁理士】

【氏名又は名称】　岡田　守弘

【電話番号】　03-3725-2215

【手数料の表示】

【予納台帳番号】　015543

【納付金額】　21,000円

【提出物件の目録】

【物件名】　明細書　1

【物件名】　図面　1

【物件名】　要約書　1

【包括委任状番号】　9705795

【プルーフの要否】　要

【書類名】　明細書

【発明の名称】　拡大鍵生成装置および記録媒体

【特許請求の範囲】

【請求項１】

暗号鍵から拡大鍵を生成する拡大鍵生成装置において、

入力された暗号鍵のビット列を複数のグループに分割し、これら分割した各グループのビット列に演算を複数ｉ回それぞれ行なって複数ｉの演算結果を生成し、これら生成した各グループ毎の複数ｉの演算結果について複数のグループ間で該当演算結果をそれぞれ１つにまとめる演算を行ない、複数ｉの中間データを生成する中間データ生成手段と、

指定された拡大鍵の段数ｒをもとに、上記複数ｉの中間データから１つを選択し、選択した中間データを非可逆変換して段数ｒの拡大鍵を生成する拡大鍵生成手段と

を備えたことを特徴とする拡大鍵生成装置。

【請求項２】

上記入力された暗号鍵のビット列として、入力された暗号鍵のビット列に非線型関数を演算したビット列としたことを特徴とする請求項１記載の拡大鍵生成装置。

【請求項３】

上記１つにまとめる演算として、論理演算としたことを特徴とする請求項１記載の拡大鍵生成装置。

【請求項４】

上記１つにまとめる演算を行なった後、非線型関数を演算して中間データを生成したことを特徴とする請求項１記載の拡大鍵生成装置。

【請求項５】

上記選択した中間データについて段数ｒに従った転置を行なった後に非可逆変換を行なうことを特徴とする請求項１記載の拡大鍵生成装置。

【請求項６】

入力された暗号鍵のビット列を複数のグループに分割し、これら分割した各グ

ループのビット列に演算を複数ｉ回それぞれ行なって複数ｉの演算結果を生成し、これら生成した各グループ毎の複数ｉの演算結果について複数のグループ間で該当演算結果をそれぞれ１つにまとめる演算を行ない、複数ｉの中間データを生成する中間データ生成手段と、

指定された拡大鍵の段数ｒをもとに、上記複数ｉの中間データから１つを選択し、選択した中間データを非可逆変換して段数ｒの拡大鍵を生成する拡大鍵生成手段と

して機能させるプログラムを記録したコンピュータ読取可能な記録媒体。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、暗号鍵から拡大鍵を生成する拡大鍵生成装置および記録媒体に関するものである。

【０００２】

【従来の技術】

一般的な共通鍵暗号による暗号化処理の構成を図８に示す。図８の暗号化装置は、平文と暗号鍵を入力とし、暗号文を出力するものであって、暗号化処理装置と拡大鍵生成装置とから構成されている。暗号化処理装置は、暗号化処理１から暗号化処理ｎの順番に、ｎ段階の処理を行なう。拡大鍵処理装置は、入力された暗号鍵をもとに、ｎ段階の暗号化処理のそれぞれで用いられる拡大鍵１から拡大鍵ｎの生成を順次行なう。この拡大鍵の生成は、重要な問題であり、高速性と安全性とが要求されている。

【０００３】

従来、高速処理可能な方式として、ＤＥＳの拡大鍵生成方式がある。このＤＥＳの拡大鍵生成方式は、図９の右側の拡大鍵生成装置に示すように、暗号鍵を入力とし、巡回シフトとビット転置のみにより拡大鍵１から拡大鍵ｎの生成を行なうため、処理が高速である。

【０００４】

また、安全性がより高い方式として、　ＭＡＲＳの拡大鍵生成方式（MARS-a

candidate cipher for AES,THE First AES Conference 1998 p1-p9)がある。

【０００５】

【発明が解決しようとする課題】

上述した前者の図９のＤＥＳの拡大鍵生成方式は、循環シフトとビット転置のみ（図９の右側の＊の部分参照）により拡大鍵の生成を行なうため、処理が高速である反面、拡大鍵のビット情報から暗号鍵のビット情報を容易に得ることができてしまうという問題がある。このため、ｎ個の拡大鍵のうち１個でも情報が漏洩した場合、暗号鍵の情報まで漏洩することとなり、安全性に問題があった。

【０００６】

また、上述した後者のＭＡＲＳの拡大鍵生成方式は、拡大鍵の情報から暗号鍵の情報を簡単に得ることができないので、安全性が高い反面、多くの演算を繰り返すため、高速に処理を行なうことができないという問題があった。

【０００７】

本発明は、これらの問題を解決するため、第１段階で暗号鍵から中間データを生成し、第２段階で中間データから任意のデータを選択して非可逆変換を行ない任意の段数の拡大鍵を生成し、任意段の拡大鍵を非可逆変換を経て高速生成して共通鍵方式の安全性を高めることを目的としている。

【０００８】

【課題を解決するための手段】

図１を参照して課題を解決するための手段を説明する。

図１において、暗号化装置１は、平文および暗号鍵を入力とし、暗号文を出力するものであって、拡大鍵処理装置３などから構成されるものである。

【０００９】

拡大鍵処理装置３は、暗号鍵から拡大鍵を生成するものであって、ここでは、中間データ生成装置４および拡大鍵生成装置５などから構成されるものである。

中間データ生成装置４は、暗号鍵を入力とし、複数ｉの中間データを生成するものである。

【００１０】

拡大鍵生成装置５は、複数ｉの中間データから指定された段数ｒの拡大鍵を生成するものである。

次に、動作を説明する。

【００１１】

中間データ生成装置４が入力された暗号鍵のビット列を複数のグループに分割し、これら分割した各グループのビット列に演算を複数回ｉそれぞれ行なって複数ｉの演算結果を生成し、これら生成した各グループ毎の複数ｉの演算結果について複数のグループ間で該当演算結果をそれぞれ１つにまとめる演算を行ない、複数ｉの中間データを生成し、拡大鍵生成装置５が指定された拡大鍵の段数ｒをもとに複数ｉの中間データから１つを選択し、選択した中間データを非可逆変換して段数ｒの拡大鍵を生成するようにしている。

【００１２】

この際、入力された暗号鍵のビット列として、入力された暗号鍵のビット列に非線型関数を演算したビット列とするようにしている。

また、１つにまとめる演算として、論理演算を行なうようにしている。

【００１３】

また、１つにまとめる演算を行なった後、非線型関数を演算して中間データを生成するようにしている。

また、選択した中間データについて段数ｒに従った転置を行なった後に非可逆変換を行なうようにしている。

【００１４】

従って、第１段階で暗号鍵から中間データを生成し、第２段階で中間データから任意のデータを選択して非可逆変換を行ない任意の段数の拡大鍵を生成することにより、拡大鍵を非可逆変換を経て高速生成して共通鍵方式の安全性を高めることが可能となる。

【００１５】

【発明の実施の形態】

次に、図１から図７を用いて本発明の実施の形態および動作を順次詳細に説明する。

【００１６】

図１は、本発明のシステム構成図を示す。

図１において、暗号化装置１は、平文および暗号鍵を入力とし、暗号文を出力するものであって、暗号化処理装置２および拡大鍵処理装置３などから構成されるものである。

【００１７】

暗号化処理装置２は、拡大鍵１から拡大鍵ｎをもとに暗号化処理（１）から暗号化処理（ｎ）のｎ段階の処理を行ない、暗号文を作成して出力するものである。暗号化処理（１）から暗号化処理（ｎ）は、拡大鍵処理装置３で生成された拡大鍵１から拡大鍵ｎを受け取ってそれぞれの暗号化処理を行ない、最終段から暗号文を出力する。

【００１８】

拡大鍵処理装置３は、暗号鍵から拡大鍵を生成するものであって、ここでは、中間データ生成装置４および拡大鍵生成装置５などから構成されるものである。

中間データ生成装置４は、暗号鍵を入力とし、複数ｉの中間データを生成するものである（図２、図３など参照）。

【００１９】

拡大鍵生成装置５は、複数ｉの中間データから指定された段数ｒの拡大鍵を生成するものである（図２、図４など参照）。

次に、図２のフローチャートの順番に従い図１の構成のもとで暗号鍵から拡大鍵を生成するときの動作を詳細に説明する。

【００２０】

図２は、本発明の動作説明フローチャートを示す。

図２において、Ｓ１は、ユーザ鍵を入力する。これは、図１の暗号化装置１に平文と一緒に、暗号鍵（ユーザ鍵）を入力する。これにより、図１の拡大鍵処理装置３を構成する中間データ生成装置４がユーザ鍵（暗号鍵）を取り込んだこととなる。

【００２１】

Ｓ２は、非線型関数Ｍを演算する。これは、後述する図３に示すように、暗号

鍵のビット列を８グループに分割し、各グループのビット列に非線型関数Ｍを演算する（図６、図７を用いて後述する）。

【００２２】

Ｓ３は、偶数のときに定数を加算する。

Ｓ４は、奇数のときに定数を乗算する。これらＳ３、Ｓ４は、後述する図３に示すように、非線型関数Ｍを演算した後のビット列に、偶数のときは定数を加算し、奇数のときは定数を乗算する。

【００２３】

Ｓ５は、排他的論理和を演算する。これは、後述する図３に示すように、Ｓ４で定数を加算したビット列と、定数を乗算したビット列とを排他的論理和の演算を行なう。

【００２４】

Ｓ６は、非線型関数Ｍを演算する。これは、Ｓ５で演算した後のビット列に、非線型関数Ｍを演算し、中間データを生成する。尚、Ｓ４、Ｓ５、Ｓ６で、複数の定数（例えば図３の場合にはｉ＝０，１，２に対応する３つ定数）についてそれぞれ加算、乗算した後に、排他的論理和演算、更に非線型関数演算をそれぞれ行ない、図３の場合にはそれぞれ３つからなる中間データを生成する。

【００２５】

以上のＳ１からＳ６によって、ユーザ鍵（暗号鍵）を入力とし、例えば図３の構成に従い、非可逆変換を経た複数づつ（図３では３つづつ）からなる中間データを作成することが可能となる。

【００２６】

図２のＳ７は、段数ｒを入力する。これは、作成しようとする、拡大鍵の段数ｒを入力する。これにより、本発明では、指定した段数の拡大鍵を以降のＳ８からＳ１１で直接に作成することが可能となる。

【００２７】

Ｓ８は、中間データから該当する値を選択する。これは、後述する図４の（ａ）の複数ｉづつの中間データ中から、段数ｒをもとに該当する中間データを１つづつそれぞれ選択する。

【００２８】

Ｓ９は、段数ｒに従った転置を行なう。これは、図４の（ａ）のデータ並び替え処理装置で、段数ｒを入力して、Ｓ８で選択した中間データの転置を行なう（図５の（ｃ），（ｄ）参照）。

【００２９】

Ｓ１０は、転置後の中間データの非可逆変換Ｇを行なう（図４の（ｂ）参照）。

Ｓ１１は、ｒ段数の拡大鍵を出力する。これは、Ｓ１０で演算した後のデータをｒ段数の拡大鍵として出力する。

【００３０】

Ｓ１２は、終わりか判別する。ＹＥＳの場合には、終了する。ＮＯの場合には、Ｓ７に戻り、次の指定された段数ｒの拡大鍵の作成を行なう。

以上のＳ７からＳ１１によって、Ｓ１からＳ６で作成した複数ｉづつの中間データをもとに、指定された段数ｒに従って１つづつの中間データを選択し、段数ｒに従って転置した後、非可逆変換を行い、指定された段数ｒの拡大鍵を非可逆変換を経て高速に生成することが可能となる。以下順次詳細に説明する。

【００３１】

図３は、本発明の説明図（中間データ）を示す。

図３において、上段のｋ０からｋ７は、暗号鍵（ユーザ鍵）のビット列を順次８分割したビット列である。

【００３２】

上段および下段のＭは、非線型関数演算を表す（後述する図６、図７参照）．

＋は、定数の加算を表す。ここで、Ｍ（４ｉ）の加算は、分割したビット列のｋの添字が例えば既述した図２のＳ３の偶数の場合に、ｉ＝０，１，２のときの定数Ｍの値をそれぞれ加算した３つを出力する旨を表す。同様に、Ｍ（４ｉ＋１）、Ｍ（４ｉ＋２）、Ｍ（４ｉ＋３）についも、ｉ＝０，１，２のときの定数Ｍの値をそれぞれ加算した３つを出力する旨を表す。

【００３３】

×は、定数の乗算を表す。ここで、ｉ＋１の乗算は、分割したビット列のｋの

添字が例えば既述した図２のＳ４の奇数の場合に、ｉ＝０，１，２のときの定数の値をそれぞれ乗算した３つを出力する旨を表す。

【００３４】

＋は、加算および乗算した結果の排他的論理和の演算を行なう旨を表す。

a_iからd_iは、ｈ個づつの中間データの出力を表す。ここでは、a_iからd_i（ｉ＝０，１，２）の３つづつの出力された中間データを表す。

【００３５】

以上のように、暗号鍵を８つのグループに分割してそれぞれのビット列に非線型関数Ｍを演算した後、偶数のものに３つの定数をそれぞれ加算および奇数のものに３つの定数を乗算し、加算および乗算した該当データを排他的論理和演算で１つにまとめ、更に非線型関数Ｍを演算して３つづつの中間データを生成することが可能となる。

【００３６】

図４は、本発明の説明図（拡張鍵）を示す。

図４の（ａ）は、ｉ個づつの中間データから段数ｒをもとに１つを選択して拡大鍵を生成するシステム構成を示す。

【００３７】

図４の（ａ）において、中間データは、ここでは、a_i、b_i，c_i，d_i（ｉ＝０，１，２）からなる、図３の構成で生成された中間データ（中間鍵）である。

【００３８】

ｓｅｌｅｃｔ値決定装置は、作成しようとする拡大鍵の段数ｒをもとに、中間データa_i、b_i，c_i，d_i（ｉ＝０，１，２）のうちのいずれのｉのものを選択するかを決定するものである。決定は、後述する図５の（ａ）の式（１）に従い決定する。

【００３９】

ｓｅｌｅｃｔｏｒは、ｓｅｌｅｃｔ値決定装置によって決定されたX_r、Y_r、Z_r、W_rに従い、ここでは、ｉ＝０，１，２の該当するものの中間データａ（X_r）、ｂ（Y_r）、ｃ（Z_r）、ｄ（W_r）を１つそれぞれ選択するもので

ある。

【００４０】

データ並び替え処理装置は、段数ｒをもとに、選択された中間データａ（X_r）、ｂ（Y_r）、ｃ（Z_r）、ｄ（W_r）の並び替え（転置）を行なうものである。この転置は、後述する図５の（ｄ）のように、段数ｒに対応した転置を行なうものである。

【００４１】

Ｇ（Ｘ、Ｙ、Ｚ、Ｗ、ｒ）計算装置は、並び替え後の中間データ（Ｘ，Ｙ，Ｚ，Ｗ）をもとに、拡大鍵$E_x Key_r$を生成するものである。これは、後述する図４の（ｂ）の構成で拡大鍵$E_x Key_r$を生成する。

【００４２】

以上の構成によって、中間データから段数ｒを指定して、当該段数ｒの拡大鍵を生成することが可能となる。以下順次詳細に説明する。

図４の（ｂ）は、図４の（ａ）のＧ（Ｘ，Ｙ，Ｚ，Ｗ）計算装置の詳細構成を示す。

【００４３】

図４の（ｂ）において、１ｂｉｔ左巡回シフトは、データのビット列を１ビット左方向に巡回してシフトするものである。

排他的論理和は、２つのデータの排他的論理和の演算を行なうものである。

【００４４】

加算は、２つのデータを加算するものである。

減算は、１つのデータから他のデータを減算するものである。

以上の回路を図示のように接続して指定段数をもとに選択および転置した後の中間データ（Ｘ，Ｙ，Ｚ，Ｗ）から拡大鍵を生成することが可能となる。

【００４５】

図５は、本発明の説明図を示す。

図５の（ａ）は、ｉ個づつの中間データから段数ｒのものを１つ選択するときに使用する式（１）を示す。図示の式（１）は下記である。

【００４６】

$X_r = Z_r = r \mod 3$

$Y_r = W_r = r + [r/3] \mod 3$

図５の（ｂ）は、図５の（ａ）の式（１）を模式的に示す。これは、図５の（ａ）の式（１）の値を実際に計算したものであって、段数ｒのときに０，１，２の３個中から１つを選択する値であって、９個で巡回するものである。

【００４７】

以上の図５の（ａ），（ｂ）で段数ｒに対応する値（ｉ＝０，１，２の３個のうちの１つ）を決定し、既述した図４の（ａ）でｉ個づつの中間データ中から当該段数ｒで決った（X_r，Y_r，Z_r，W_r）を選択することが可能となる。

【００４８】

図５の（ｃ）は、オーダ表を示す。このオーダ表は、図５の（ａ），（ｂ）で選択した段数ｒの中間データ（X_r，Y_r，Z_r，W_r）を並び換える（置換）するときのオーダ（順序）を決定するものである。ここでは、左側の段数ｒに対応づけて右側の並び替え順序に示すように、並び替える（図４の（ａ）のデータ並び替え処理装置が実行する）。

【００４９】

次に、図６および図７を用いて非線型関数演算について説明する。

図６の（ａ）は、非線型関数Ｍの演算の全体の構成の例を示す。ここでは、ユーザ鍵（暗号鍵）ｍ（例えば３２ビット）を入力とし、非線型関数Ｍを演算して結果ｗ（３２ビット）を生成するときの様子を説明する。

【００５０】

（１）　ユーザ鍵の３２ビットをここでは図示のように、６，５，５，５，５６ビットで順次ｍ０，ｍ１，ｍ２，ｍ３，ｍ４，ｍ５にそれぞれ分割する。

（２）　５ビットに分割したｍ１，ｍ２，ｍ３，ｍ４は、図６の（ｂ）のＳ５（ｘ）の表に従い、該当するｘに対応するＳ５（ｘ）の値にそれぞれ変換する。

【００５１】

（３）　同様に、６ビットに分割したｍ０，ｍ６は、図６の（ｃ）のＳ６（ｘ）の表に従い、該当するｘに対応するＳ６（ｘ）の値にそれぞれ変換する。

（４）　（２）と（３）によってｖが図６の（ａ）のデータｖが生成されたこ

ととなる。

【００５２】

（５）　図７の（ｅ）に模式的に示す行列式に、図７の（ｄ）から矢印で示す位置に置き、次に、（４）で生成したデータｖを置き、両者の行列演算を行ない、右側のｗを計算する。これにより、図６の（ａ）のＭＤＳを用いたＸＯＲ計算装置による結果（非線型関数Ｍの演算結果）が得られたこととなる。

【００５３】

以上によって、図６の（ａ）の構成に示す順番に従い、例えば図３の非線型関数Ｍの演算を行なうことが可能となる。

図６の（ｂ）は図６の（ａ）のＳ５（ｘ）のテーブル例を示し、図６の（ｃ）は図６の（ａ）のＳ６（ｘ）のテーブル例を示す。

【００５４】

図７の（ｃ）は非線型関数Ｍを行列演算するときの定数（ＭＤＳを用いてＸＯＲ計算装置で用いる定数）を示し、図７の（ｄ）はＭＤＳを用いたＸＯＲ計算装置で行なう行列演算を模式的に示す。

【００５５】

次に、既述したユーザ鍵から中間データを生成する第１段階の処理、および中間データから指定された段数ｒの拡大鍵を生成する第２段階の処理について数式，記号を用いて説明する。

【００５６】

（１）　第１段階の処理（ユーザ鍵から中間データを生成する処理）：

（１－１）２５６ビットの暗号鍵を、３２ビットごとに８個のデータｋ０，ｋ１，・・・ｋ７に分割する（図３参照）

（１－２）　（１－１）で分割した３２ビットを入力とし、３２ビットを出力する非線型関数Ｍを用いて、以下の（１－３）から（１－６）の計算により、中間データａ（ｉ），ｂ（ｉ），ｃ（ｉ），ｄ（ｉ）の生成をｉ＝０，１，２について行なう（図３参照）。また、非線型関数Ｍについて、（３－１）から（３－６）を実行する。

【００５７】

（１－３）　ａ（ｉ）＝Ｍ（Ｔａ（ｋ０，ｉ）ＸＯＲ　Ｕａ（ｋ１，ｉ）を計算する。ただし、Ｔａ（ｋ０，ｉ）＝Ｍ（ｋ０）＋Ｍ（４ｉ），Ｕａ（ｋ１，ｉ）＝Ｍ（ｋ１）×（ｉ＋１）である。尚、ＸＯＲは、排他的論理和演算を表す。

【００５８】

（１－４）　ｂ（ｉ）＝Ｍ（Ｔｂ（ｋ２，ｉ）　ＸＯＲ　Ｕｂ（ｋ３，ｉ）を計算する。ただし、Ｔｂ（ｋ３，ｉ）＝Ｍ（ｋ２）＋Ｍ（４ｉ＋１），Ｕｂ（ｋ３，ｉ）＝Ｍ（ｋ３）×（ｉ＋１）である。

【００５９】

（１－５）　ｃ（ｉ）＝Ｍ（Ｔｃ（ｋ４，ｉ）　ＸＯＲ　Ｕｃ（ｋ５，ｉ）を計算する。ただし、Ｔｃ（ｋ４，ｉ）＝Ｍ（ｋ４）＋Ｍ（４ｉ＋２），Ｕｃ（ｋ５，ｉ）＝Ｍ（ｋ５）×（ｉ＋１）である。

【００６０】

（１－６）ｄ（ｉ）＝Ｍ（Ｔｄ（ｋ６，ｉ）　ＸＯＲ　Ｕｄ（ｋ７，１））を計算する。ただし、Ｔｄ（ｋ６，ｉ）＝Ｍ（ｋ６）＋Ｍ（４ｉ＋３），Ｕｄ（ｋ７，ｉ）＝Ｍ（ｋ７）×（ｉ＋１）である。

【００６１】

（２）　第２段階の処理（中間データから指定された段数ｒの拡大鍵を生成する処理）：

（２－１）ｒ段数の拡大鍵ＥｘＫｅｙｒ（ｒ＝０．１・・）について、以下の（２－２）から（２－４）に従い計算を行なう（図４の（ａ）参照）。

【００６２】

（２－２）　Ｘｒ＝Ｚｒ＝ｒ　ｍｏｄ３，Ｙｒ＝Ｗｒ＝ｒ＋[ｒ／３]ｍｏｄ３（式（１））で表される数列Ｘ，Ｙ，Ｚ，Ｗを用いて、（Ｘ，Ｙ，Ｚ，Ｗ）＝（ａ（Ｘｒ），ｂ（Ｙｒ），ｃ（Ｚｒ），ｄ（Ｗｒ））とする。

【００６３】

（２－３）　ｒ’＝（ｒ＋[ｒ／３６]）ｍｏｄ１２を満たすｒ’に関して、（Ｘ、Ｙ、Ｚ、Ｗ）＝ＯＲＤＥＲ＿１２（Ｘ，Ｙ，Ｚ，Ｗ，ｒ’）で示されるデータ並び替えを行なう。ただし、ＯＲＤＥＲ＿１２（Ｘ，Ｙ，Ｚ，Ｗ，ｒ’）は、図５の（ｃ）に従う。

【００６４】

（２－４）　ＥｘＫｅｙｒ＝Ｇ（Ｘ，Ｙ，Ｚ，Ｗ）により、ｒ段数の拡大鍵を計算する。ただし、Ｇ（Ｘ，Ｙ，Ｚ，Ｗ）＝（（ｘ＜＜＜１）＋Ｙ）　ＸＯＲ（（（Ｚ＜＜＜１）－Ｗ）＜＜＜１）であり、＜＜＜１は１ビット左巡回シフトを表す（図４の（ｂ）参照）。

【００６５】

（３）　非線型関数Ｍの演算処理：

（３－１）　３２ビット入力ｍから、以下の（３－２）から（３－６）に従い、３２ビットのｗを出力する（図６の（ａ）参照）。

【００６６】

（３－２）　ｍをビット分割した値ｍ０，・・・ｍ５を、以下に従って与える。

ｍ０＝（ｍの第０ビット目から第５ビット）

ｍ１＝（ｍの第６ビット目から第１０ビット）

ｍ２＝（ｍの第１１ビット目から第１５ビット）

ｍ３＝（ｍの第１６ビット目から第２０ビット）

ｍ４＝（ｍの第２１ビット目から第２５ビット）

ｍ５＝（ｍの第２６ビット目から第３１ビット）

（３－３）　５ビットの入力に対し５ビットを出力する非線型変形関数Ｓ５，６ビットの入力に対し６ビットを出力する非線型変換関数Ｓ６を用いて、

ｓ０＝Ｓ６（ｍ０）

ｓ１＝Ｓ５（ｍ１）

ｓ２＝Ｓ５（ｍ２）

ｓ３＝Ｓ５（ｍ３）

ｓ４＝Ｓ５（ｍ４）

ｓ５＝Ｓ６（ｍ５）

ここで、Ｓ５、Ｓ６を既述した図６の（ｂ）、（ｃ）にそれぞれ示す。

【００６７】

（３－４）　ｖ＝ｓ０｜ｓ１｜ｓ２｜ｓ３｜ｓ４｜ｓ５を計算する。｜は、ビ

ット値の連結を表す。

（３－５）　ｖのｉ番目のビット値ｖｉ，および５ビット入力から３２ビットを出力する変換テーブルＭＤＳを用いて、

ｗ＝（ｖ０×ＭＤＳ（０））ＸＯＲ（ｖ１×ＭＤＳ（１））ＸＯＲ・・・ＸＯＲ（ｖ３１×ＭＤＳ（３１））を計算する。ただし、ｖｉ×ＭＤＳ（ｉ）は、ｖｉ＝０のとき０、ｖｉ＝１のときＭＤＳ（ｉ）である。また、ＭＤＳは図７の（ｄ）に従う。

【００６８】

（３－６）　ｗを出力する。

【００６９】

【発明の効果】

以上説明したように、本発明によれば、第１段階で暗号鍵から中間データを生成し、第２段階で中間データから任意のデータを選択して非可逆変換を行ない任意の段数の拡大鍵を生成する構成を採用しているため、拡大鍵を非可逆変換を経て高速生成して共通鍵方式の安全性を高めることが可能となる。これにより、

（１）　例えば中間データを１個生成するためには大きな処理時間が必要であるあるが、拡大鍵生成装置５により、必要な中間データの個数を少なくでき、安全性の高い拡大鍵を高速に生成できる。

【００７０】

（２）　また、生成した拡大鍵ＥｘＫｅｙ０，ＥｘＫｅｙ１，・・・ＥｘＫｅｙｎ－１の全てを記憶せずに、暗号化あるいは復号化の処理の途中で、必要となる拡大鍵のみを生成する場合、指定した段数ｒの拡大鍵のみを高速生成できるという顕著な特徴がある。これの効果を以下説明する。

【００７１】

一般的に共通鍵暗号方式は、暗号化でＥｘＫｅｙ０，ＥｘＫｅｙ１，・・・ＥｘＫｅｙｎ－１の順に拡大鍵が用いられる場合、復号化では、ＥｘＫｅｙｎ－１・・・ＥｘＫｅｙ１，ＥｘＫｅｙ０のように、暗号化とは逆の順で拡大鍵が用いられる。ここで、ＥｘＫｅｙ１を生成するのにＥｘＫｅｙ０の値を必要とする拡大鍵生成方式を用いて（既述した図９参照）、逐次生成を行なった場合、Ｅｘ

Ｋｅｙ１は直接生成することができず、先にＥｘＫｅｙ０を生成してこれを用いてＥｘＫｅｙ１を生成することとなり、その分、復号化の拡大鍵生成時間が暗号化よりも遅くなる。

【００７２】

これに対し、本発明は、他の拡大鍵とは独立して、任意の段数ｒを指定して拡大鍵を生成できるため、拡大鍵をＥｘＫｅｙ０，ＥｘＫｅｙ１，・・・ＥｘＫｅｙｎ－１の順に生成するのも、ＥｘＫｅｙｎ－１・・・ＥｘＫｅｙ１，ＥｘＫｅｙ０の順に生成するのも、同じ処理時間で行なうことができるという顕著な特徴がある。

【００７３】

以上のように、本発明では、拡大鍵の逐次生成を行なう場合でも、暗号化と復号化の処理時間を同じにして、復号化の拡大鍵生成時間を暗号化よりも遅くなることを防ぐ顕著な効果がある。

【図面の簡単な説明】

【図１】

本発明のシステム構成図である。

【図２】

本発明の動作説明フローチャートである。

【図３】

本発明の説明図（中間データ）である。

【図４】

本発明の説明図（拡大鍵）である。

【図５】

本発明の説明図である。

【図６】

本発明の説明図（非線型関数演算、その１）である。

【図７】

本発明の説明図（非線型関数演算、その２）である。

【図８】

共通鍵方式の説明図である。

【図９】

従来のＤＥＳのアルゴリズムの全体のブロック図である。

【符号の説明】

１：暗号化装置

２：暗号化処理装置

３：拡大鍵処理装置

４：中間データ生成装置

５；拡大鍵生成装置

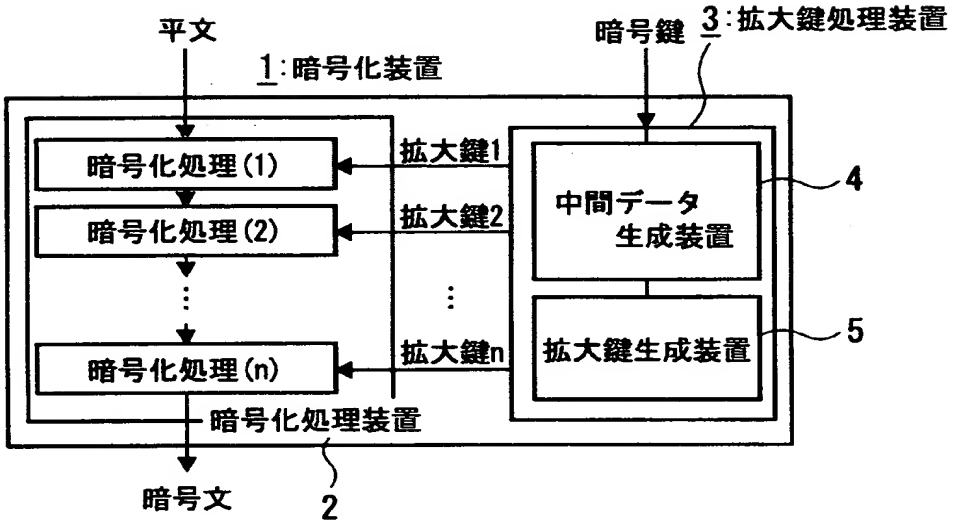
Ｍ：非線型関数

Ｇ：非可逆関数

【書類名】　　図面

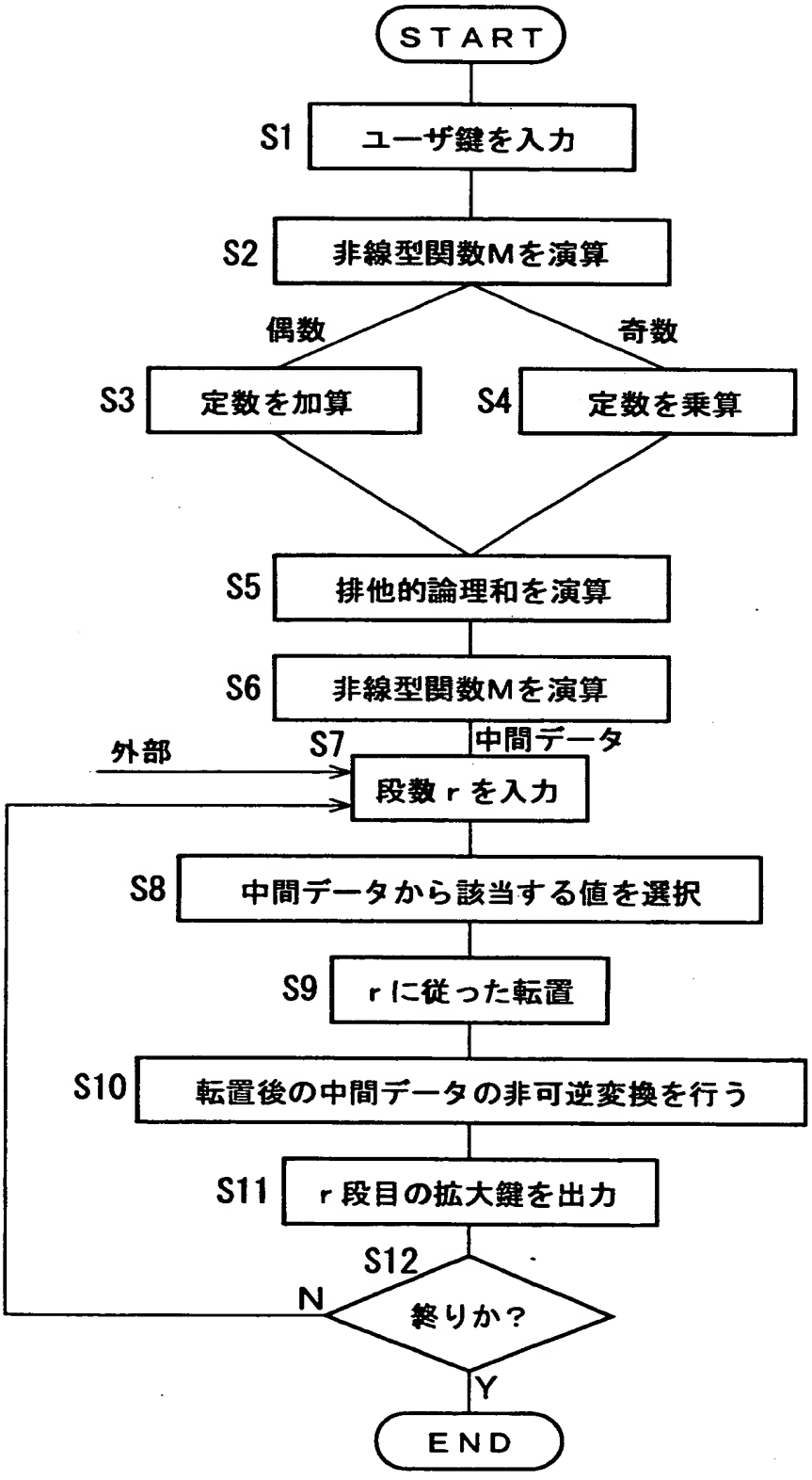
【図１】

本発明のシステム構成図

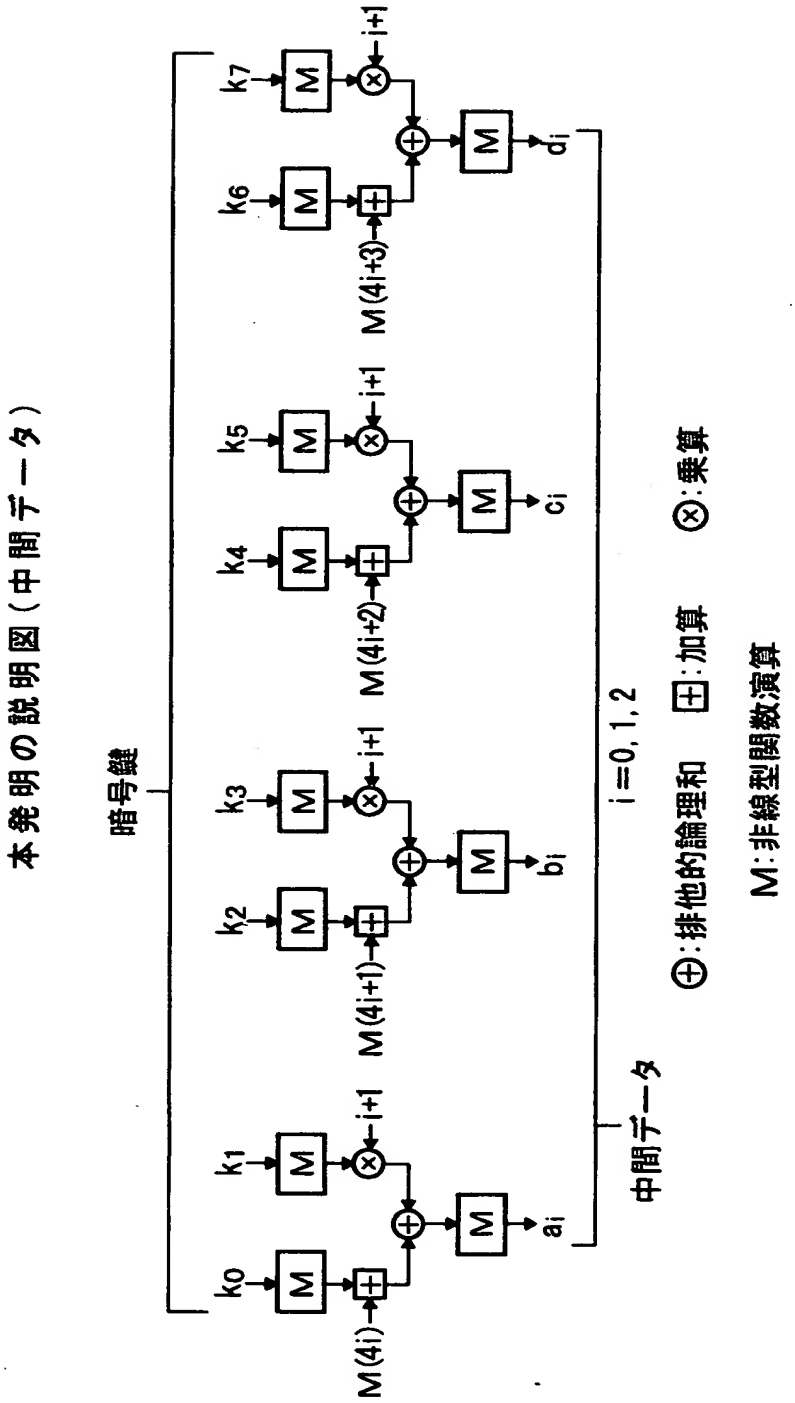


【図２】

本発明の動作説明フローチャート

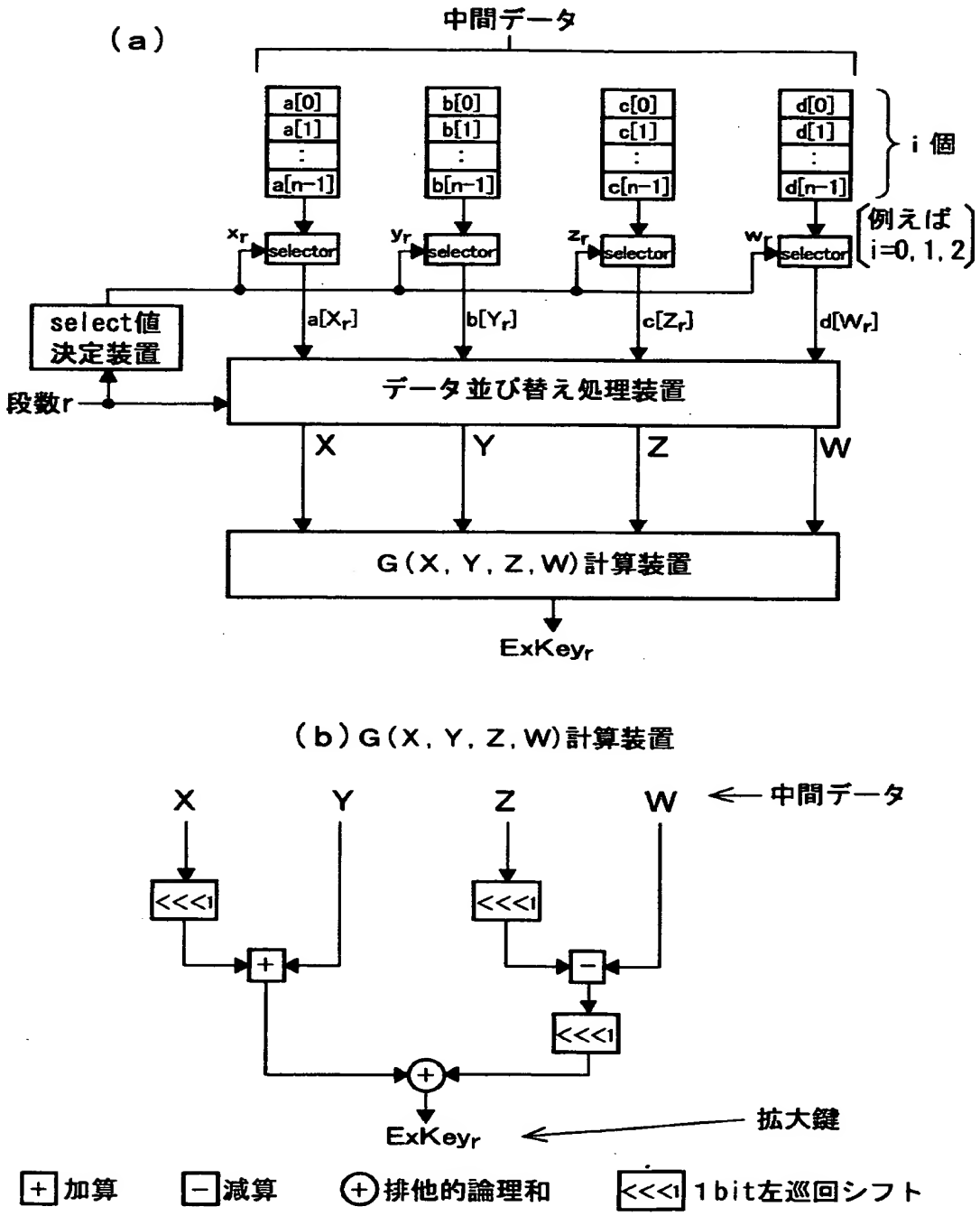


【図３】


本発明の説明図（中間データ）
暗号鍵
k0
k1
k2
k3
k4
k5
k6
k7
M
M(4i)
M(4i+1)
M(4i+2)
M(4i+3)
i+1
ai
bi
ci
di
中間データ
i=0, 1, 2
⊕:排他的論理和
⊞:加算
⊗:乗算
M:非線型関数演算

【図４】

本発明の説明図（拡大鍵）



【図５】

本発明の説明図

（ａ）$x_r = z_r = r \bmod 3$, $y_r = w_r = r + [r/3] \bmod 3$・・・式（１）

（ｂ）

９個で巡回

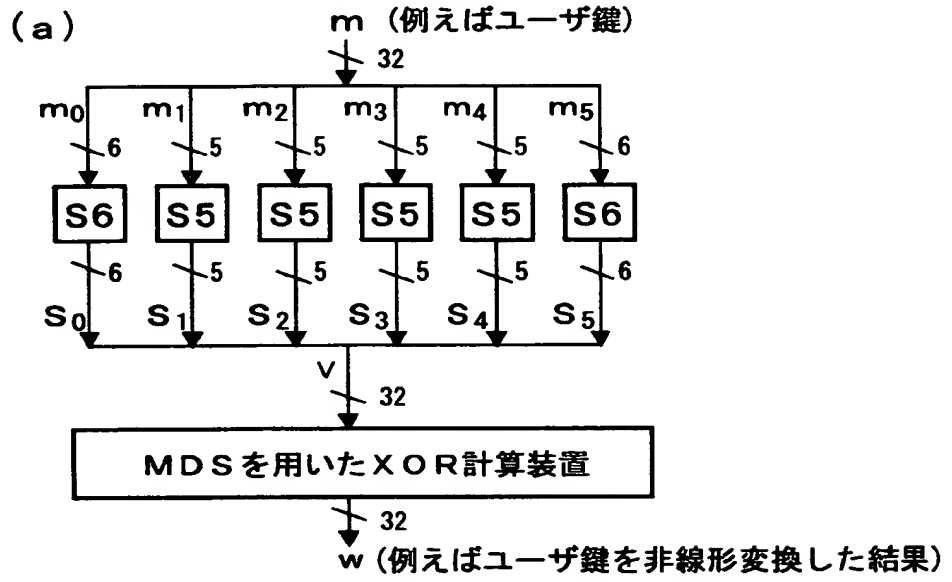
r	0	1	2	3	4	5	6	7	8	9	0. . .
X_r	0	1	2	0	1	2	0	1	2	0	1. . .
Y_r	0	1	2	1	2	0	2	0	1	0	1. . .
Z_r	0	1	2	0	1	2	0	1	2	0	1. . .
W_r	0	1	2	1	2	0	2	0	1	0	1. . .

（ｃ）オーダ表

r（段目）	ORDER_12(X,Y,Z,W,r) ← 並び替え
0	(X,Y,Z,W)
1	(Y,X,W,Z)
2	(Z,W,X,Y)
3	(W,Z,Y,X)
4	(X,Z,W,Y)
5	(Y,W,Z,X)
6	(Z,X,Y,W)
7	(W,Y,X,Z)
8	(X,W,Y,Z)
9	(Y,Z,X,W)
１０	(Z,Y,W,X)
１１	(W,X,Z,Y)

【図６】

本発明の説明図（非線型関数演算．その１）



（ｂ）Ｓ５（ｘ）

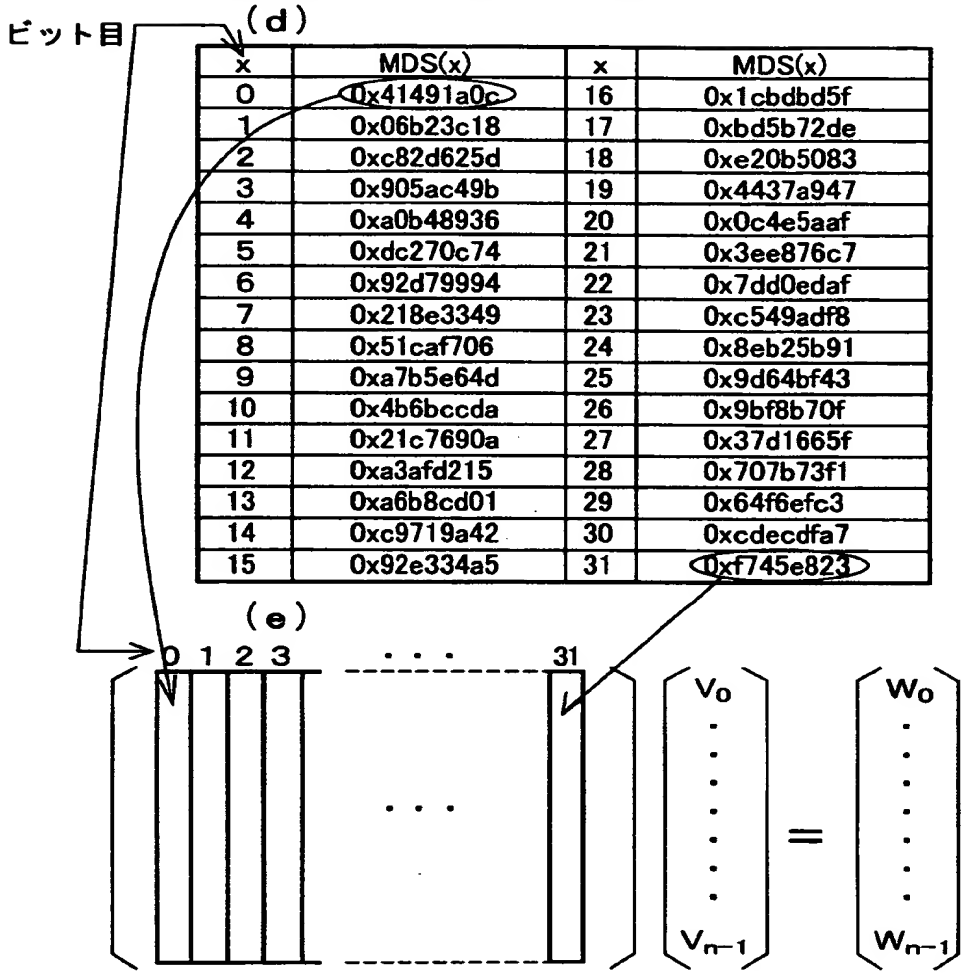
x	S5(x)	x	S5(x)	x	S5(x)	x	S5(x)
0	20	8	22	16	27	24	23
1	26	9	30	17	11	25	5
2	7	10	13	18	1	26	8
3	31	11	14	19	21	27	3
4	19	12	4	20	6	28	0
5	12	13	24	21	16	29	17
6	10	14	9	22	2	30	29
7	15	15	18	23	28	31	25

（ｃ）Ｓ６（ｘ）

x	S6(x)	x	S6(x)	x	S6(x)	x	S6(x)
0	47	16	37	32	62	48	3
1	59	17	63	33	52	49	16
2	25	18	20	34	35	50	41
3	42	19	61	35	18	51	34
4	15	20	55	36	14	52	33
5	23	21	2	37	46	53	7
6	28	22	30	38	0	54	45
7	39	23	44	39	54	55	49
8	26	24	9	40	17	56	50
9	38	25	10	41	40	57	58
10	36	26	6	42	27	58	1
11	19	27	22	43	4	59	21
12	60	28	53	44	31	60	43
13	24	29	48	45	8	61	57
14	29	30	51	46	5	62	32
15	56	31	11	47	12	63	18

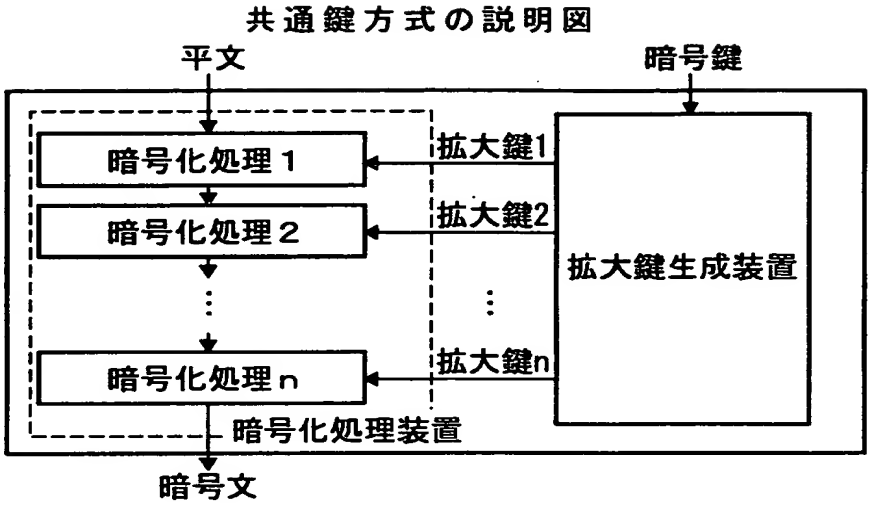
【図７】

本発明の説明図（非線型関数演算．その２）



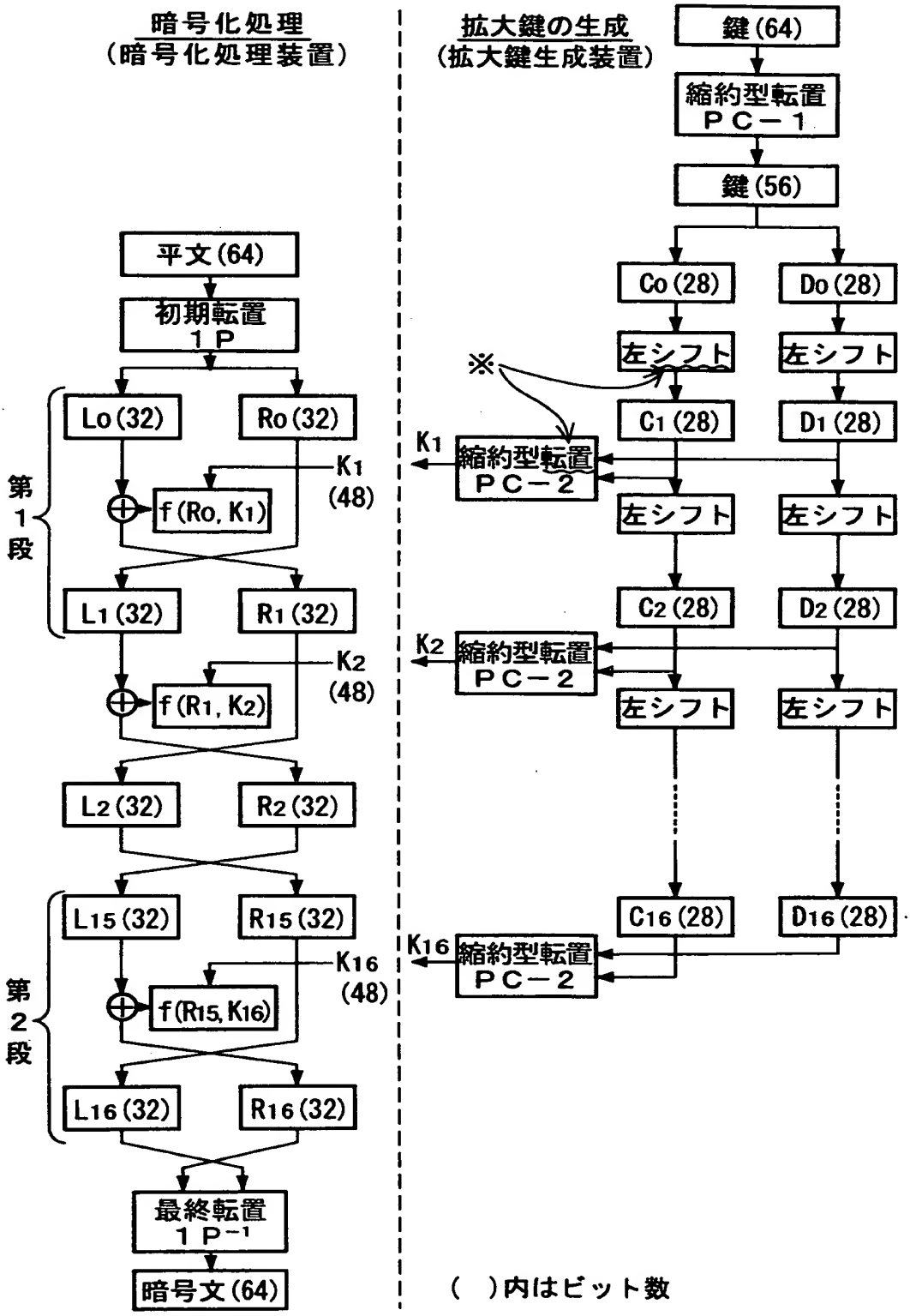
x	MDS(x)	x	MDS(x)
0	0x41491a0c	16	0x1cbdbd5f
1	0x06b23c18	17	0xbd5b72de
2	0xc82d625d	18	0xe20b5083
3	0x905ac49b	19	0x4437a947
4	0xa0b48936	20	0x0c4e5aaf
5	0xdc270c74	21	0x3ee876c7
6	0x92d79994	22	0x7dd0edaf
7	0x218e3349	23	0xc549adf8
8	0x51caf706	24	0x8eb25b91
9	0xa7b5e64d	25	0x9d64bf43
10	0x4b6bccda	26	0x9bf8b70f
11	0x21c7690a	27	0x37d1665f
12	0xa3afd215	28	0x707b73f1
13	0xa6b8cd01	29	0x64f6efc3
14	0xc9719a42	30	0xcdecdfa7
15	0x92e334a5	31	0xf745e823

【図８】


共通鍵方式の説明図
平文
暗号鍵
暗号化処理１
暗号化処理２
暗号化処理ｎ
拡大鍵1
拡大鍵2
拡大鍵n
拡大鍵生成装置
暗号化処理装置
暗号文

【図９】

従来のＤＥＳのアルゴリズムの全体のブロック図



【書類名】　要約書

【要約】

【課題】　本発明は、暗号鍵から拡大鍵を生成する拡大鍵生成装置および記録媒体に関し、第１段階で暗号鍵から中間データを生成し、第２段階で中間データから任意のデータを選択して非可逆変換を行ない任意の段数の拡大鍵を生成し、任意段の拡大鍵を非可逆変換を経て高速生成して共通鍵方式の安全性を高めることを目的とする。

【解決手段】　入力された暗号鍵のビット列を複数のグループに分割し、これら分割した各グループのビット列に演算を複数ｉ回それぞれ行なって複数ｉの演算結果を生成し、これら生成した各グループ毎の複数ｉの演算結果について複数のグループ間で該当演算結果をそれぞれ１つにまとめる演算を行ない、複数ｉの中間データを生成する中間データ生成手段と、指定された拡大鍵の段数ｒをもとに、複数ｉの中間データから１つを選択し、選択した中間データを非可逆変換して段数ｒの拡大鍵を生成する拡大鍵生成手段とを備えるように構成する。

【選択図】　図１

出　願　人　履　歴　情　報

識別番号　　〔０００００５２２３〕

１．変更年月日　　１９９６年　３月２６日
［変更理由］　住所変更
住　所　神奈川県川崎市中原区上小田中４丁目１番１号
氏　名　富士通株式会社